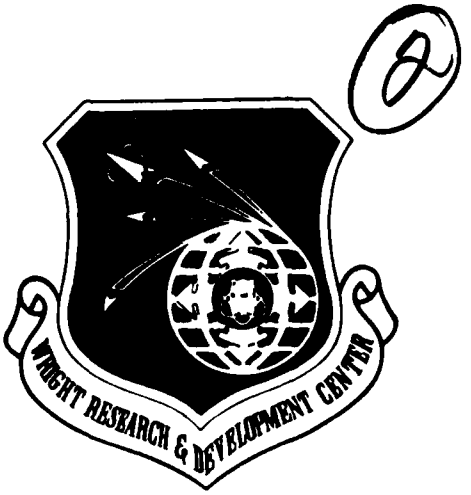


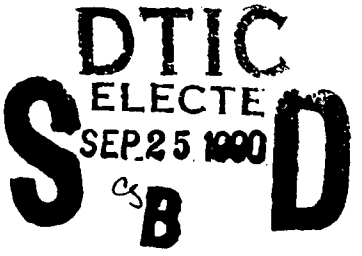
WRDC-TR-90-2014

THE FEASIBILITY OF USING NEURAL NETWORKS AND OTHER OPTIMIZATION ALGORITHMS TO OBTAIN CROSS SECTIONS FROM ELECTRON SWARM DATA

Kinema Research
18720 Autumn Way
Monument CO 80132

April 1990

Final Report For Period June 1989 - October 1989



AERO PROPULSION AND POWER LABORATORY
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6563

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Alan Garscadden

ALAN GARSCADDEN

FOR THE COMMANDER

Lowell D. Massie

LOWELL D. MASSIE
Power Components Branch
Aerospace Power Division
Aero Propulsion and Power Laboratory

Michael D Braydich

MICHAEL D. BRAYDICH, Maj, USAF
Actg Chief, Aerospace Power Division
Aero Propulsion & Power Laboratory

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/POOC, WPAFB, OH 45433-6563 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS: N/A
2a. SECURITY CLASSIFICATION AUTHORITY: N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for Public Release; Distribution is Unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S):	5. MONITORING ORGANIZATION REPORT NUMBER(S): WRDC TR-90-2014
6a. NAME OF PERFORMING ORGANIZATION: Kinema Research (6b. OFFICE SYMBOL (If applicable))	7a. NAME OF MONITORING ORGANIZATION: Wright Research and Development Center Aero Propulsion and Power Laboratory
6c. ADDRESS (City, State, and ZIP Code): 18720 Autumn Way Monument, CO 80132	7b. ADDRESS (City, State, and ZIP Code): Wright-Patterson AFB, OH 45433-6563
8a. NAME OF FUNDING/SPONSORING ORGANIZATION (8b. OFFICE SYMBOL (If applicable))	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: F33615-86-C-2171
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.
61102F	2301	S2	03

11. TITLE (Include Security Classification)
The Feasibility of Using Neural Networks and Other Optimization Algorithms to Obtain Cross Sections from Electron Swarm Data.

12. PERSONAL AUTHOR(S)
W. Lowell Morgan

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 6/89 TO 10/89	April 1990	42

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Draft velocity, cross section, numerical optization, simplex method.
20	5	1	
20	6	2	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Three kinds of numerical optimization algorithms have been investigated for use in estimating the electron mimentum transfer and excitation cross sections for atoms and molecules based on measured electron transport, or swarm data. The methods investigated are the downhill or creeping simplex; simulated annealing; and neural networks. These methods have been used to obtain the cross section for momentum transfer for a model system from E/N (Electric field, E, divided by the total gas density, N) dependent drift velocities and characteristic energies. In addition the creeping simplex has been used to obtain momentum transfer cross sections for He and Ar and the momentum transfer cross section and a vibrational excitation cross section for methane from measured drift volocity and characteristic energy data. A neural network has been used to obtain an estimate of the momentum transfer cross section of xenon in the vicinity of the Ramsauer minimum from swarm data. These results serve as examples of what may be possible using these and, perhaps other optimization algorithms.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL: Dr Alan Garscadden	22b. TELEPHONE (Include Area Code): (513) 255-2923	22c. OFFICE SYMBOL: WRDC/POOC-3

DD Form 1473, JUN 86 *Previous editions are obsolete.*

PREFACE

This research was performed by W. Lowell Morgan in the period from May 15, 1989 through September 30, 1989 with the support of the Wright Research and Development Center, Wright–Patterson Air Force Base, Ohio through Subcontract CB415–21 (Task No. 15) from the San Jose State University Foundation (Prime Contract No. F33615–86–C–2171). The Air Force contract manager was Dr. Alan Garscadden, Wright R & D Center, who has had a long standing interest in this kind of approach to swarm studies [see A. Garscadden, "Current applications and opportunities in swarm studies," in Ref. 4 of this report] and who suggested this problem. Most of the calculations were performed on a 25 MHz 80386/87 personal computer. The electron swarm and cross section data used in this research came from the atomic data base of the Atomic Collisions Data Center at the Joint Institute for Laboratory Astrophysics of the University of Colorado and the National Institute of Standards and Technology.

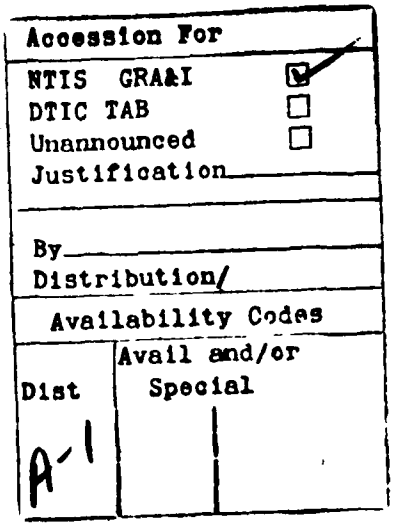


TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

The use of theory to obtain collision cross sections from electron transport data, one of the "inverse problems" of physics, was pioneered by Townsend and by Ramsauer in the 1920's. The method used in such early analyses involved measuring the drift velocity of electrons in a gas as a function of E/p (electric field strength divided by gas pressure) and inverting the integral relating the drift velocity and the momentum transfer cross section using an approximate expression for the energy distribution of the electrons. This technique has increased in sophistication over the years. In the 1960's Phelps and various collaborators applied electronic computation to the problem and developed algorithms for solving the Boltzmann equation for transport of electrons in a weakly ionized plasma to obtain an accurate electron energy distribution function valid at higher fields and in the presence of inelastic and, even, superelastic collisions. This began an era that has given us very accurate momentum transfer and lower energy (rotational and vibrational) inelastic cross sections that have been derived from measurements of the drift and diffusion of electrons in gases. This methodology is reviewed in Refs. [1–4]. This has become an increasingly active field in recent years due, for example, to the desire for cross sectional data on molecules such as CH_4, CF_4, SF_6, SiH_4, and SiF_4 that are used in semiconductor plasma processing and in switching applications.

Clearly the iterative process of choosing energy dependences of test cross sections; solving the Boltzmann equation for a range of values of electric field; computing transport coefficients; comparing to measured values; revising the test cross sections; etc. is very labor intensive and "hands on." It is obviously a process where the experience of the researcher plays an important role comparable to that of the specific computational techniques used. The object of the research being reported on here is to evaluate several computational methods for reducing the labor involved in this deconvolution process.

SECTION II

BOLTZMANN'S EQUATION AND ELECTRON SWARMS

The pertinent equations in this problem are the electron Boltzmann equation and its energy integral, which relates the various transport coefficients for electrons in a gas. We can see how the various aspects of this problem are related to each other by examining the so called two–term expansion of the Boltzmann equation and the various electron transport coefficients. If we take the general form for the Boltzmann equation,

$$(\partial/\partial t + \mathbf{v}\cdot\nabla_r + \frac{e\mathbf{E}}{m}\cdot\nabla_v)f(\mathbf{r},\mathbf{v},t) = (\partial f/\partial t)_{\text{collisions}}$$

neglect spatial and temporal dependence of the distribution function $f(\mathbf{r},\mathbf{v},t)$, and express $f=f(\mathbf{v})$ as the first two terms of a spherical harmonic expansion, that is

$$f(\mathbf{v}) = f_0(v) + \frac{\mathbf{v}}{v}\cdot\mathbf{f}_1(v)$$

then we obtain the following scalar equation for $f_0(\epsilon)$ (where $\epsilon=mv^2/2$):

$$\frac{1}{3}(eE/N)^2\, d/d\epsilon\{\epsilon/\sigma_m\, df_0/d\epsilon\} + d/d\epsilon\{(2m\sigma_m/M)\epsilon^2[f_0(\epsilon) + kT\, df_0/d\epsilon]\}$$
$$+\sum_i [(\epsilon+\epsilon_i)\sigma_i(\epsilon+\epsilon_i)f_0(\epsilon+\epsilon_i) - \epsilon\sigma_i(\epsilon)f_0(\epsilon)] = 0 \qquad (1)$$

Here we have assumed that the populations of the excited levels, labeled by i, are small enough that superelastic collisions and transitions among excited states are unimportant. The electron impact cross sections involved are σ_m, the momentum transfer cross section, and $\{\sigma_i\}$, the set of cross sections for transitions from the ground state to the various excited states $\{i\}$. This equation does a remarkably good job of describing the transport of electrons under the influence of an electric field in most gases.

The Boltzmann equation, via the probability density function $f_0(\epsilon)$, is a microscopic description of the behavior of electrons in a gas. We need to relate $f_0(\epsilon)$ to some macroscopic quantities that can be measured. This is done by performing an energy integral of (1), which gives the following:

$$v_d\, E/N - \sqrt{2e/m}\int \epsilon^2 (2m/M)\sigma_m(\epsilon)[f_0(\epsilon)+kT\cdot df_0/d\epsilon]d\epsilon = \sum_i \epsilon_i k_i \qquad (2)$$

The drift velocity times the electric field divided by the gas number density is the power input; the second term is the power lost due to elastic collisions (which is reduced by recoil); and the RHS of the equation is the power lost due to inelastic collisions. The drift velocity, v_d, and the elastic collision term contain the momentum transfer cross section, while all the terms involve integrals over the electron energy distribution function, $f_0(\epsilon)$, itself a function of the cross sections. The two most commonly measured transport coefficients are the drift velocity, v_d, and the tranverse diffusion coefficient, D_T, which are related to $f_0(\epsilon)$ and the momentum transfer cross section, $\sigma_m(\epsilon)$ by the following:

$$v_d \approx \int[\sigma_m(\epsilon)]^{-1}(df_0/d\epsilon)\epsilon d\epsilon \quad \text{and} \quad D_T \approx \int[\sigma_m(\epsilon)]^{-1} f_0(\epsilon)\epsilon d\epsilon$$

We see that the drift velocity and diffusion coefficent sample different aspects of $f(\epsilon)$ and, hence, represent two somewhat independent pieces of information. Generally the quantity D_T/μ, the *characteristic energy*, is reported in the literature, rather than D_T itself. For a Maxwellian distribution of electrons, where $f_0(\epsilon) \approx \exp(-\epsilon/kT_e)$, for which an electron temperature, T_e, can be defined, the Einstein relation $D_T/\mu = kT_e = 2<\epsilon>/3$ holds. Since the mean electron energy $<\epsilon>$ is not a measureable quantity (it is usually computed by solving Boltzmann's equation), the characteristic energy is generally the only measure of

electron energy that we have. We see that comparison with measured D_T/μ values gives us another constraint on the cross sections as does comparison with measured rate coefficients, $k_i \approx \int \sigma_i(\epsilon) f_0(\epsilon) \epsilon d\epsilon$, and spectral data where they are available.

The relationship between the cross sections and the transport coefficients via the distribution function $f_0(\epsilon)$ is highly nonlinear. We have a mapping

$$\left\{ \begin{array}{c} \sigma_m(\epsilon) \\ \\ \{\sigma_i(\epsilon)\} \end{array} \right\} \quad ======> \quad \left\{ \begin{array}{c} v_d(E/N) \\ D/\mu(E/N) \\ \{k_i(E/N)\} \end{array} \right\}$$

and we want to find the reverse mapping given the transport data. It has been claimed in the literature (see Ref. 5, for example) that the reverse mapping is not unique, but we have never seen it proven. It seems likely that the more transport data we have available, the more likely it is that the reverse mapping is going to be unique.

SECTION III

TECHNIQUES FOR RECOVERING CROSS SECTIONS FROM SWARM DATA

There are numerous techniques [6–8] that might be used in inverting electron transport data to obtain a collision cross section. The three classes of methods that we discuss here are (A) the *downhill* or *creeping simplex* algorithm, which is a topological approach; (B) function minimization by *simulated annealing*, a statistical approach; and (C) *neural networks*, which do not fit into any of the usual categories for numerical algorithms. The latter are very new and largely unknown in applications to physical problems. Descriptions of these approaches follow below. Initially we applied all three methods to a model problem in order to develop the algorithms and codes.

Another approach to solving inverse and so–called "missing information" problems is the maximum entropy method. This is a method of statistical inference that provides a least biased estimate based upon given information. Using the information theoretical definition of entropy (which it is our desire to maximize) one sets up a likelihood function [7] that is a linear combination of the entropy function and contraints, which are modified by Lagrange multipliers. Such a constraint may be the goodness–of–fit criterion, for example. One then maximizes the entropy subject to the constraints. This then yields a transcendental equation that, in principle, can be iterated upon to yield an estimate of the unknown function. Examples of the use of this approach to astrophysical problems, intermolecular potentials in solid state physics, and signal analysis can be found in Refs. [7,9,10]. We believe that the maximum entropy condition is implicit in this problem through the use of the Boltzmann equation, which is the equation that maps the cross sections into the transport coefficients. The equilibrium and steady state solutions of the Boltzmann equation are, of course, maximum entropy solutions as can be seen from the behavior of the H–function, where $dH/dt \leq 0$ by Boltzmann's H–theorem. The entropy (at equilibrium, of course) is directly related to the H–function via $H = -S/kV$ and, consequently, is maximized.

The two obvious examples to use as models are those of electrons drifting in a gas having either constant collision frequency or constant cross section. Here the electron energy distributions are Maxwell–Boltzmann and Druyvesteyn, respectively, with easily calculable E/N dependent drift velocities. Any method should be able to use the $v_d(E/N)$ curve and the appropriate $f_0(\epsilon)$ and recover $\sigma \approx 1/\sqrt{\epsilon}$ or $\sigma \approx \sigma_0$ respectively for these two special cases.

We have investigated the capability of these optimization algorithms to reproduce the constant collision frequency cross section, $\sigma = \sigma_0/\sqrt{\epsilon}$, using the drift velocity, $v_d(E/N)$, and characteristic energies, $D/\mu(E/N)$, associated with that cross section. The resulting electron energy distribution function, $f_0(\epsilon)$, v_d, D/μ, and $<\epsilon>$ are all analytic functions.[11]

A. THE CREEPING SIMPLEX

This is a very versatile method for optimization problems. In finding the minimum of a function of n variables, $F(x_1,...,x_n)$, we can think of the set $\{x_i\}$ as defining an n–dimensional surface in a space of n+1 dimensions. n+1 points on this surface then define what is called a *simplex*. If one draws a picture of the surface defined by $F(x_1,x_2)$, as shown in Fig. 1, it is easy to see that this simplex is a triangle, i.e., three points determine the two lines that define a plane in three dimensions. Now, using several transformation rules this simplex, or n–dimensional plane, can be made to move around on the surface and, specifically, can be made to follow the contours of the surface moving ever "downward" toward the lowest point. This algorithm was first published by Nelder and Mead[12] but Press, et al.[6] give a good description of it.

My implementation of the creeping simplex involves defining the initial simplex by choosing n+1 trial cross sections of the form $\sigma(\epsilon)=\sigma_0/\epsilon^p$, where the constant σ_0 is chosen from a uniform distribution of random numbers in $10^{-17}<\sigma_0<10^{-14}$ cm^2 and the power p is chosen from uniform random numbers in $0<p<1$. Using these cross sections the

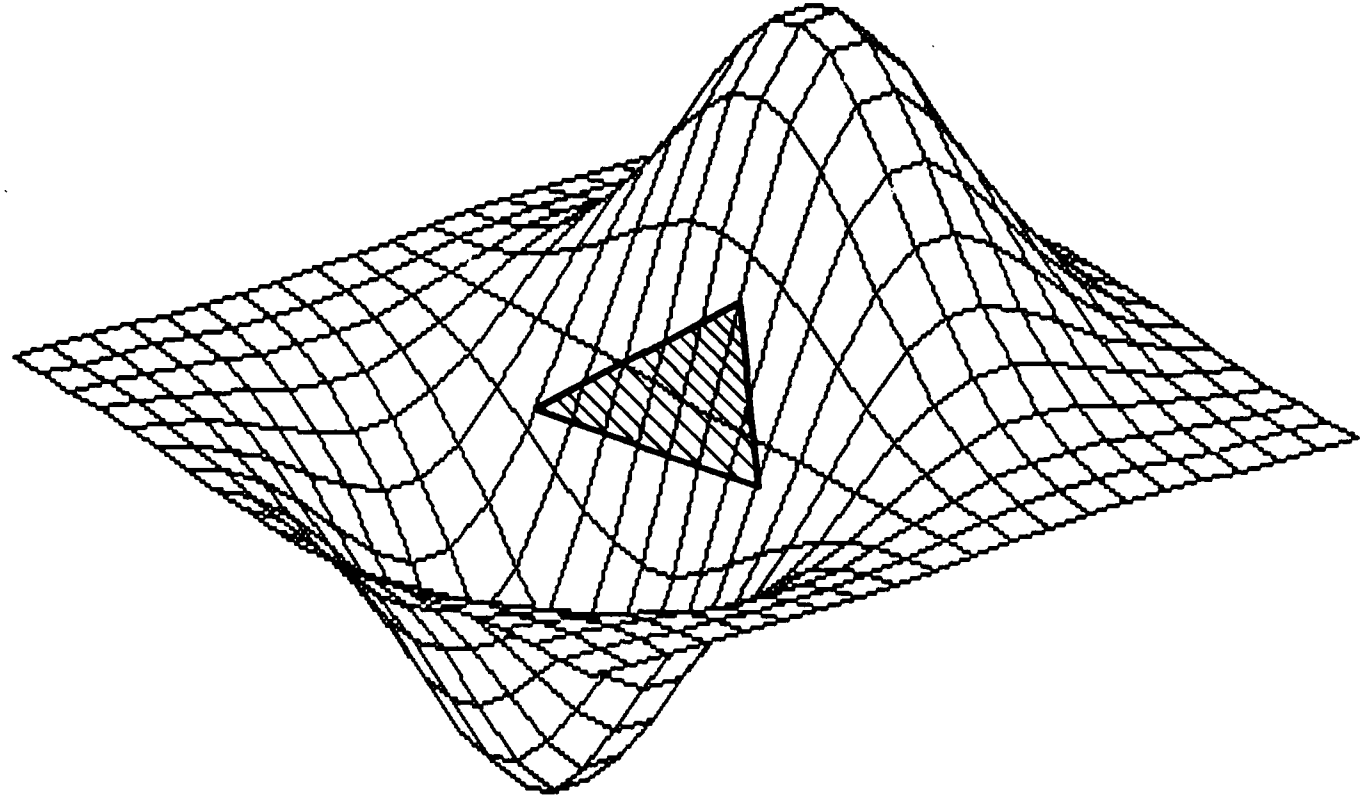


Figure 1: Example of a 2-dimensional simplex
on a 3-dimensional surface

appropriate $f(\epsilon)$'s are computed and, consequently, the set of drift velocities, w(E/N), for a number of values of E/N. The function to be minimized then is a χ^2 on the difference between the "data" $v_d(E/N)$ and the n+1 trial w(E/N), that is, $\Sigma\,[(v_d^2 - w^2)/v_d^2]$. With good convergence properties, the simplex reduces nearly to a point on the surface, so that the final result is the cross section associated with the final simplex. In Fig. 2 we see the results of using this procedure on the test problem. This calculation, which took about 20 minutes on a 25 MHz 80386 PC, used 5 values of E/N (1–10 Td) and 13 cross section points running from 0.01 eV in powers of 2. We see that the results look excellent except at the largest energies, where $\sigma_m(\epsilon)$ is insensitive to the range of E/N used.

Application to Real Gases

We have used the simplex algorithm to recover the momentum transfer cross sections for He, Ar, and CH_4 from the their E/N dependent transport coefficients. The results are shown in Figs. 3 through 6.

The He calculations used the drift velocity and characteristic energy measurements given in the book by Huxley and Crompton.[3] Eleven values each of $v_d(E/N)$ and $D/\mu(E/N)$ were used with 0.1 Td $\leq$ E/N $\leq$ 3 Td. In the calculations on real gases, the power parameter in the trial cross sections was in range $-1<p<+1$, rather than in (0,1) as in the model calculation. The orginal transport data are cited by Huxley and Crompton as coming from Refs.13 and 14. Shown for comparison is the He momentum transfer cross section of Crompton, et al.[15] Note that this cross section was derived from different drift velocity data (at 77 K rather than 300 K) and without use of D/μ data, so we expect the resulting cross section to be somewhat different from that which would be derived from the data used here. We see reasonable agreement except at the extremes in energy. This, as has been discussed above, is due to the limited range of E/N used. These calculations take about 2 hours on an 80386 PC. Better accuracy over the energy range would likely be achieved with a larger number of values of $v_d(E/N)$ and $D/\mu(E/N)$.

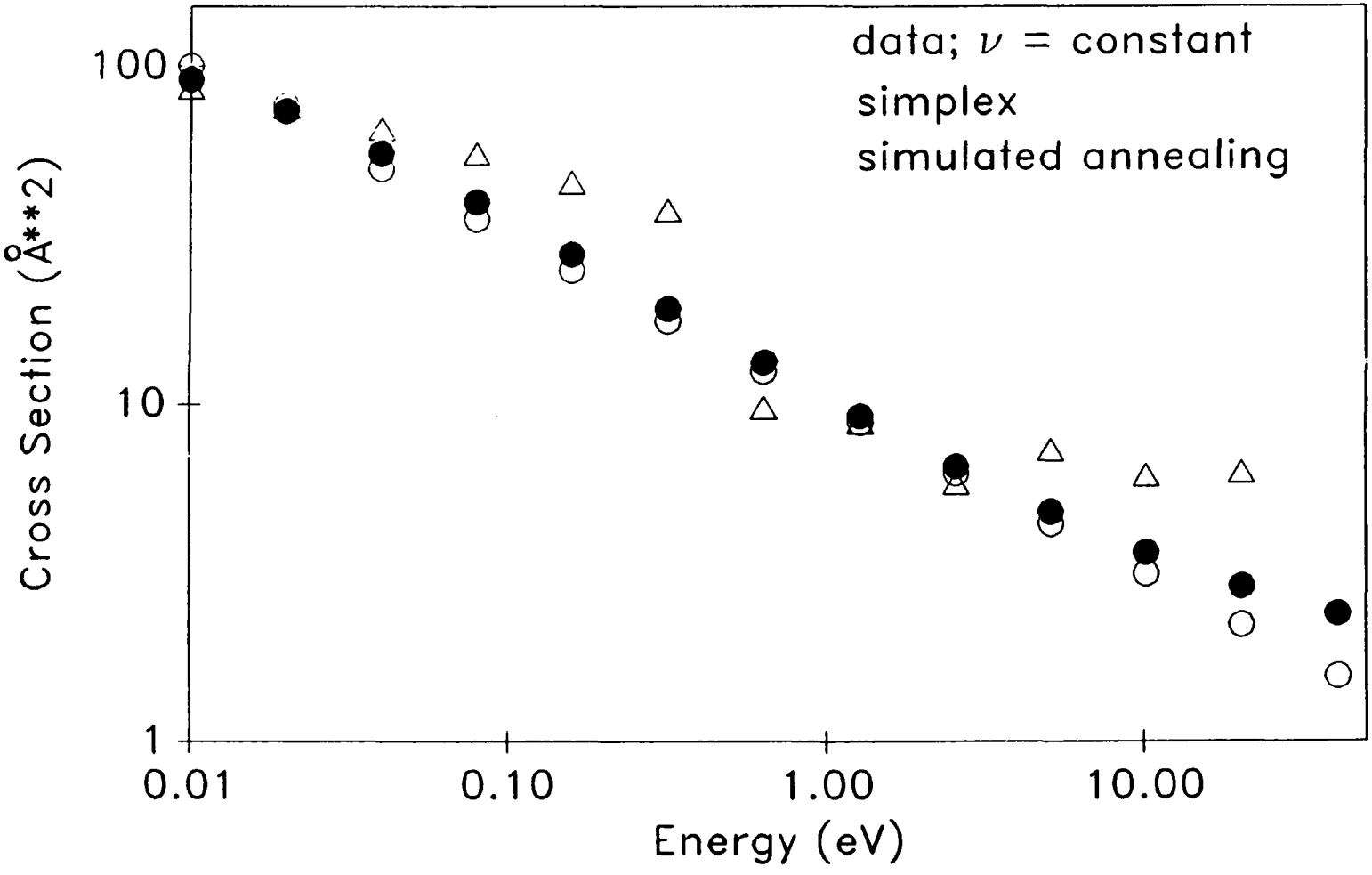


Figure 2: Momentum transfer cross section from drift velocity data for creeping simplex and simulated annealing algorithms

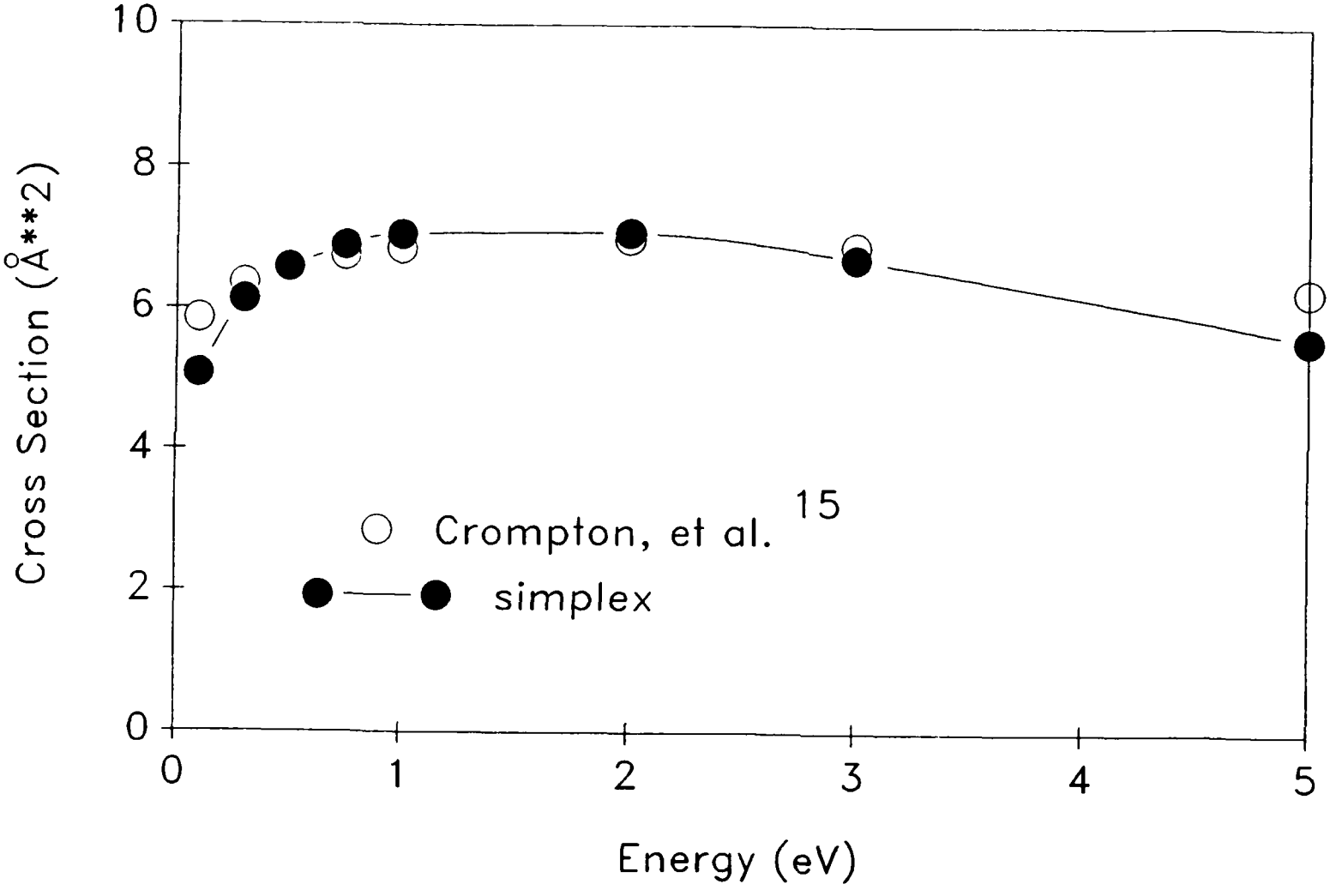


Figure 3: Downhill simplex results for He

A similar calculation performed for Ar is shown in Fig. 4. Twelve values of E/N were used in the range 0.002 Td $\leq$ E/N $\leq$ 0.1 Td, which are quite low E/N values. The 90 K transport data came from Ref. 16 (v_d) and Ref. 17 (D/μ), as cited by Huxley and Crompton. Shown for comparison are the momentum transfer cross sections for Ar found by Frost and Phelps [18] and by Milloy, Crompton, et al.[19] Neither of these cross sections was derived from the transport coefficients used in this calculation, so we expect there to be differences. This calculation is meant only to indicate the possibilities of what might be achievable with further development of this approach. Interestingly, we see that the $\sigma_m(\epsilon)$ from the optimization algorithm agrees with Milloy, et al. below the Ramsauer minimum and with Frost and Phelps above the minimum. At high energy we have the ususal problem, which would be taken care of by more E/N values over a larger range.

The most sophisticated calculation was on methane, CH_4. Methane has, in addition to a Ramsauer minimum in $\sigma_m(\epsilon)$, low energy inelastic cross sections, i.e., vibrational levels with energy losses of 0.162 and 0.361 eV. We used the same approach as described above with 12 values of E/N in the range from 0.1 to 12 Td and consistent v_d(E/N) and D/μ(E/N) data as measured by Haddad.[20] The momentum transfer cross section was computed at 12 energy values and the vibrational cross section at 10. Since this work is developmental in nature, We used only one vibrational state (0.162 eV) so that $\sigma_v(\epsilon)$ approximately represents the sum of the two actual vibrational cross sections. This approach has been used in other analyses, notably those of Pollock [21] and of Pitchford, et al.[22] We are still using the two–term expansion of the Boltzmann equation even though it is known that there is a loss of accuracy for methane and other gases that, in one way or another, do not totally satisfy the conditions used in the development of this approximation. The results of using the creeping simplex on CH_4, which required 3 hours of computational time on the PC, are shown in Figs. 5 and 6 where we see $\sigma_m(\epsilon)$ and $\sigma_v(\epsilon)$ respectively along with numerous cross sections derived by other authors from swarm

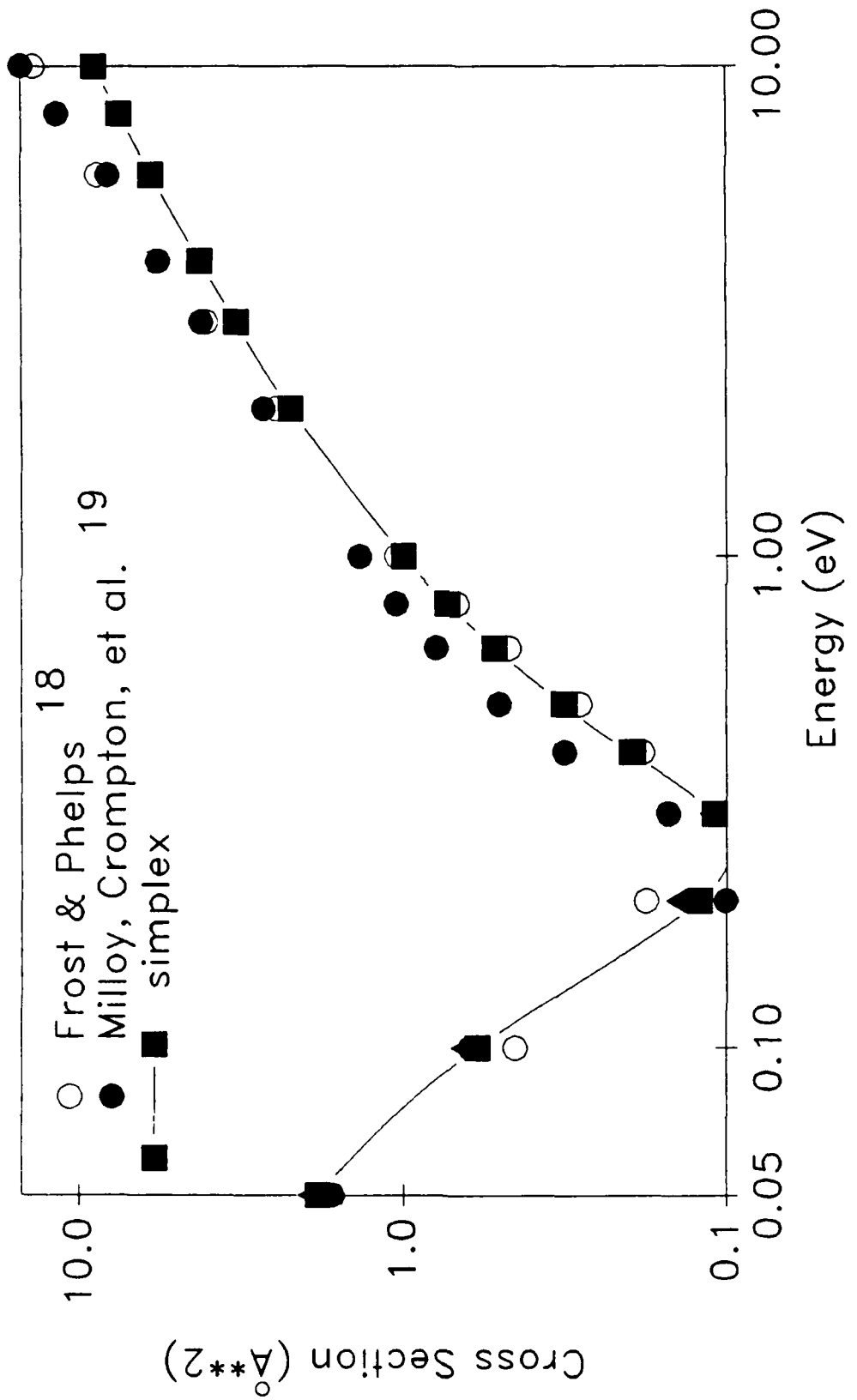


Figure 4: Downhill simplex results for Argon

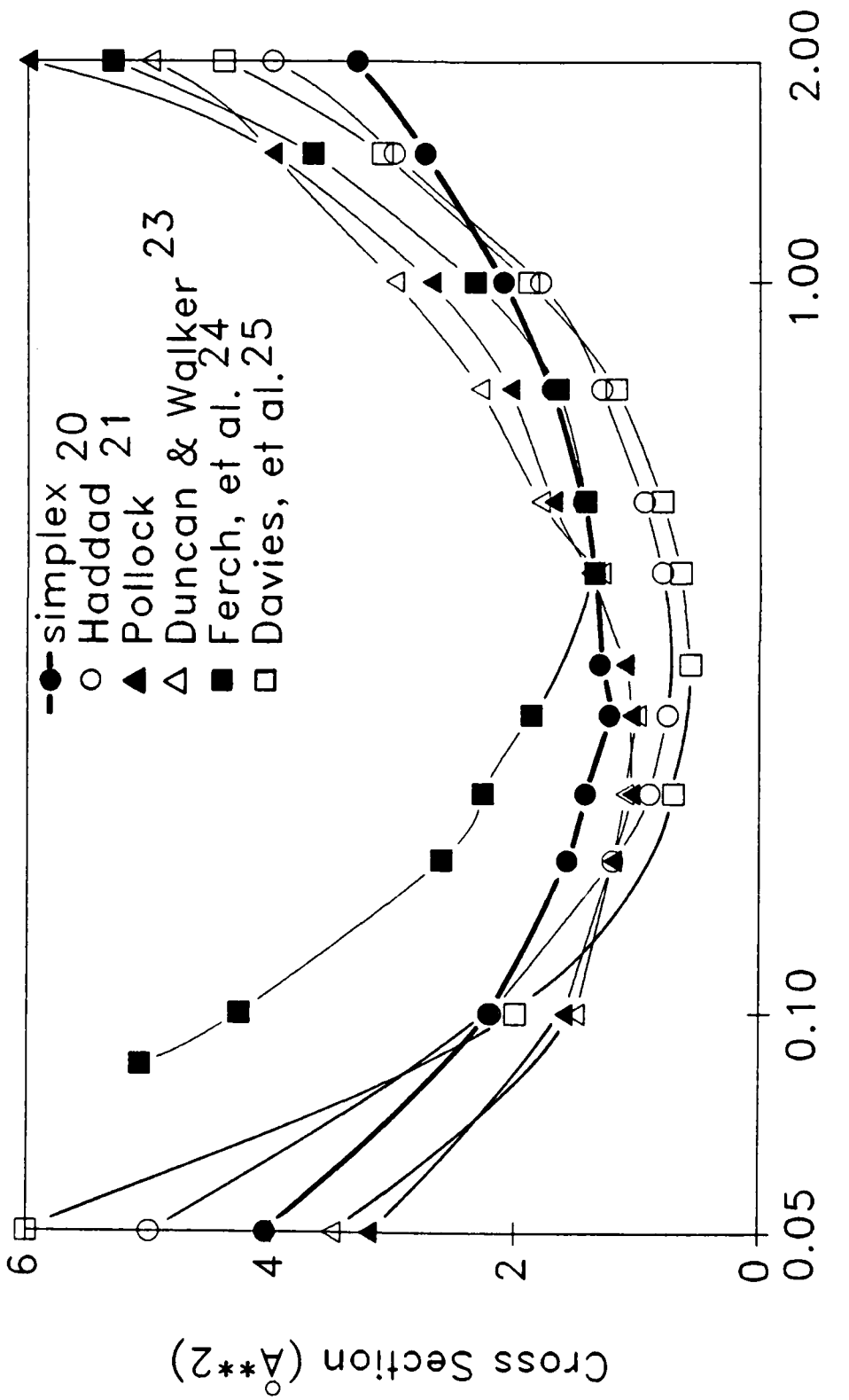


Figure 5: CH_4 momentum transfer cross sections

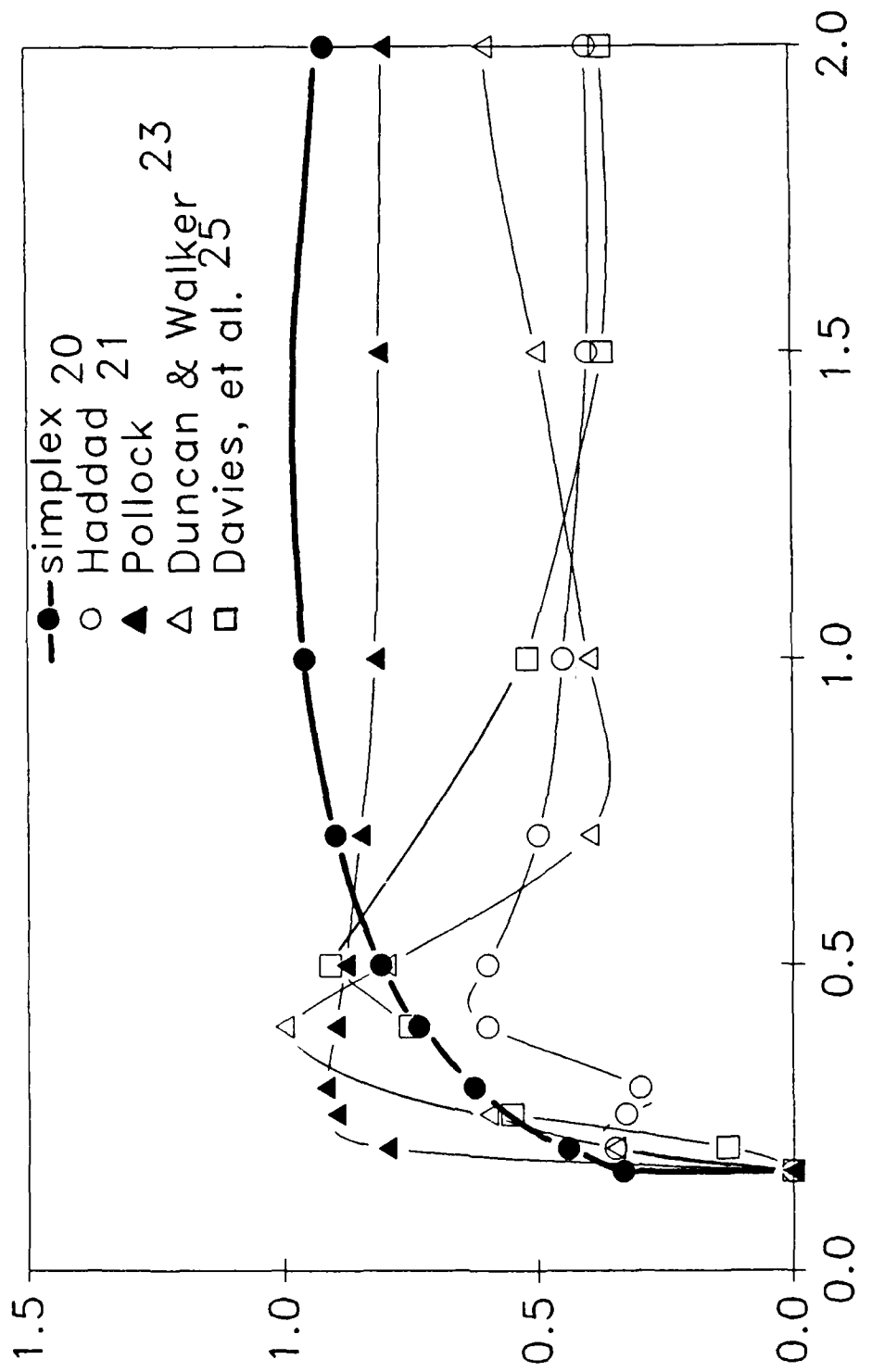


Figure 6: CH_4 vibrational cross section

data.[23–25] The first observation that we make is that there is anything but unanimity in what constitutes the "best" or "correct" cross sections. Clearly the creeping simplex results are as valid as any of the others that we see in the Figures. We see from the plots of drift velocities and characteristic energies respectively in Figs. 7 and 8 that these cross sections yield satisfactory transport coefficients. Clearly the next step in this development would be to extend this algorithm to multiple inelastic process, such as two vibrational levels in CH_4 or, more ambitiously, H_2 with rotational and vibrational levels.

B. SIMULATED ANNEALING

Simulated annealing [6,7,26,27] is a function minimization method that is an outgrowth of the Metropolis algorithm [28] commonly used for computer simulation of canonical ensembles in statistical mechanics. One minimizes a quantity E by making random changes in the configuration of the system and deciding whether or not to accept the new configurations based on comparison with the Boltzmann probability $P(E) = \exp(-E/kT)$, where T is a control parameter. When applying the method to a thermodynamic system, say a collection of atoms at temperature T, one displaces an atom at random (the Monte Carlo move) and computes the total energy E of the system. If it has decreased, the move is accepted and another MC move is made. If it has increased, the move is accepted with probability $e^{-\Delta E/kT}$. This allows the system a means of moving out of a local minimum if it has settled into one. The annealing part of the method involves slowly decreasing T as the simulation proceeds. This approach is applicable to very large systems and has been very successful in providing near optimal solutions to the so–called "traveling salesman problem."

We designed the following algorithm to perform the Monte Carlo moves appropriate to this problem. We have values of cross sections, $\sigma_j = \sigma_m(\epsilon_j)$, at some number of values of energy, ϵ_j. In the calculation presented here there are 13 energy points, i.e., $1 \le j \le 13$.

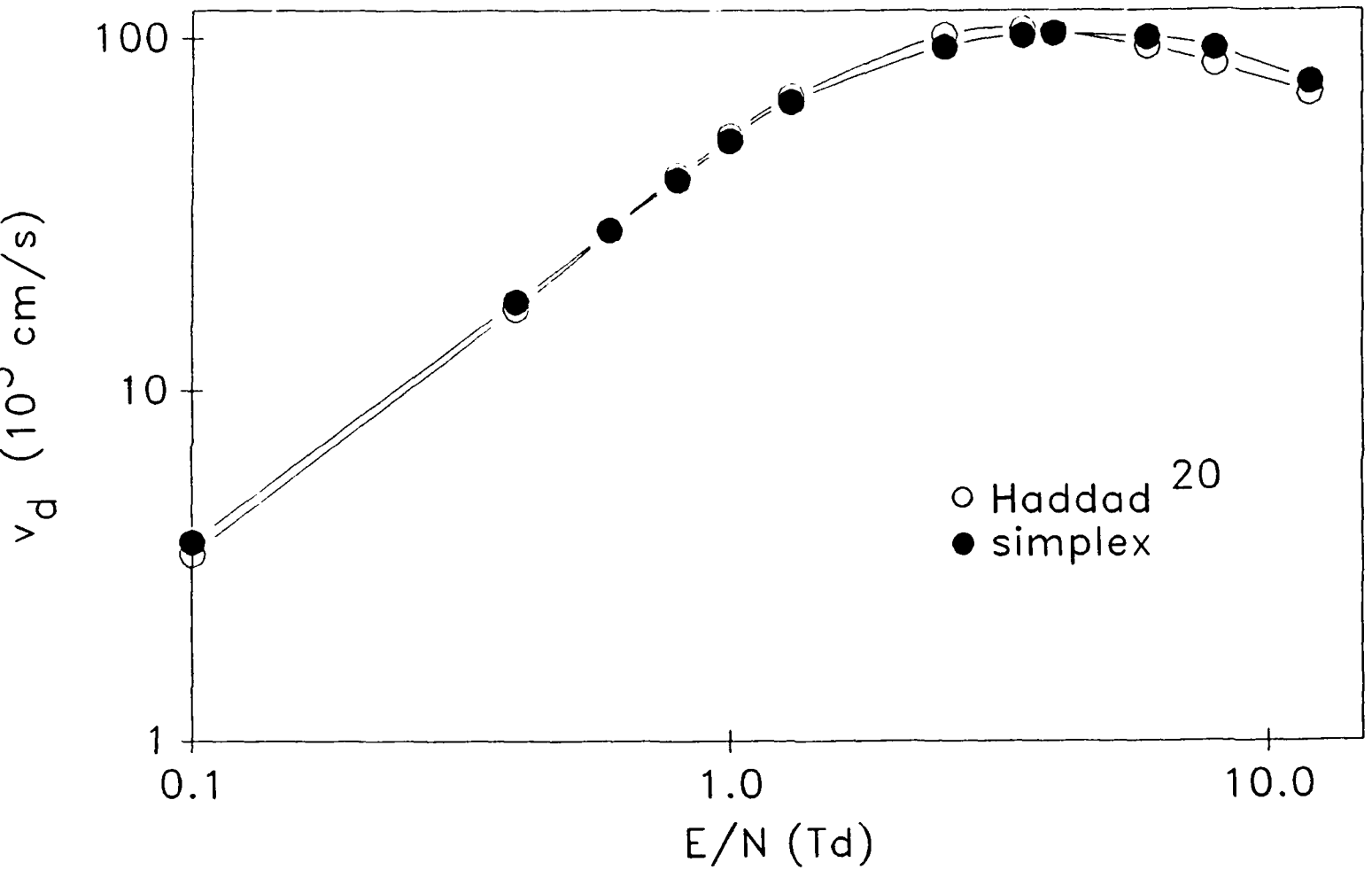


Figure 7: Drift velocities in CH_4

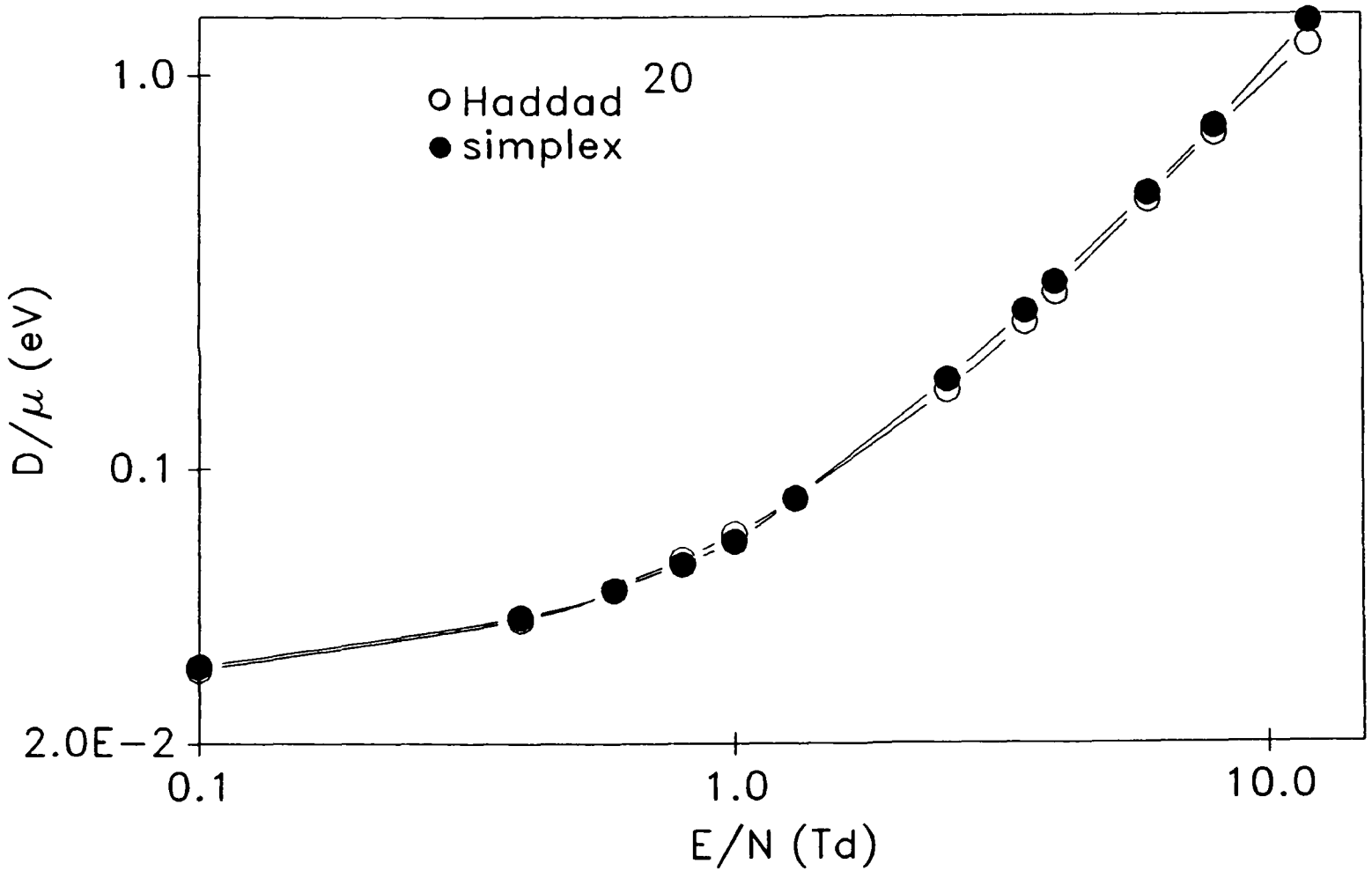


Figure 8: Characteristic energies in CH_4

object of the algorithm is to vary these in the manner described above in order to minimize the same χ^2 function that was used in the creeping simplex method. To perform this variation a j is chosen using a random number generator and then σ_j is multiplied by some scale factor. Let F^0_σ be the maximum fractional change that we allow the program to make in a σ_j in a single MC move and let η be a uniform random number in (0,1), then $\sigma_j^{i+1} = \sigma_j^i \cdot [1 + (2 \cdot \eta \cdot F^0_\sigma - F^0_\sigma)]$ is the formula for obtaining the desired random variation of σ_j from iteration to interation. In order to allow a smooth variation of the function $\sigma_m(\epsilon)$ we devised a "rubber ruler" algorithm by which σ_j is varied as just described but smaller variations are made simultaneously in the other cross sections lying nearby in energy. The complete MC move algorithm is

$$\sigma_k^{i+1} = \sigma_k^i \cdot \{1 + (2\eta F^0_\sigma - F^0_\sigma) \cdot e^{-[(\epsilon_j - \epsilon_k)/\epsilon_s]^2}\}$$

where j is chosen as above and ϵ_s is a scale energy, which was 0.2 eV is the calculations to be described.

We have applied this algorithm to the test problem described above with the results that are shown in Fig. 2. In this calculation the variations in the trial cross section were made as described above. In addition, the "temperature" kT = 0.005 and the annealing schedule was such that kT was multiplied by 0.5 after 130 successful MC moves had been made; this was carried on for 50 iterations. This particular calculation took 25 minutes of CPU time on a Cray X/MP. Although these results are not as spectacular as those obtained with the simplex, we believe this method is worth further study. The choice of the appropriate T and its annealing schedule is very much a matter of trial and error and experience [27] and this sample calculation is certainly not optimized. In addition, it is easy to see how to implement this algorithm for any number of elastic and inelastic processes, and how to use prior information on the uncertainty associated with each cross section in

performing the MC variations. Therefore, this method is likely to work better for complicated problems than is the creeping simplex algorithm, in which it is not clear how to include the latter kind of constraint.

C. NEURAL NETWORKS

This is a very new area of research.[29–31] Neural networks, which consist of layers of simulated "neurons" with associated activation functions, transfer functions, and weighting functions for the "synapse" connections to other neurons, have been shown[8,29,32,33] to be capable of computing decisions in optimization problems. Such networks have a "learning" capability in that the weights associated with connections between pairs of neurons can be modified (strengthened or weakened) in response to the network's successes and failures so as to optimize in favor of the network's successful strategies. This is probably the most novel, but least well defined, approach to the physical problem of inverting electron transport data. Aarts and Korst [34] have found that on some graph problems the neural network approach is from 20 to 400 times faster than the simulated annealing method described above.

One kind of neural network consists of a network of layers of simulated neurons as shown in Fig. 9 (taken from Ref. 31) The key elements are an input layer, one or more "hidden" layers, and an output layer. Each neuron has a transfer function associated with it that gives an output value that is some non–linear function of the sum of the input values and each pair of neurons has a weight value associated with it. The concept behind this kind of network (*feed–forward, back–propagation*) is that it can "learn" to associate a set of output patterns with a set of input patterns by adjusting the weights that connect together the network of non–linear devices. The usual transfer function used in such networks is the sigmoid $T(x)=1/(1+e^{-x})$ [there is an equivalent arctan function also]. If the output of the j^{th} neuron is o_j and w_{ij} is the weight connecting neurons i and j, then the

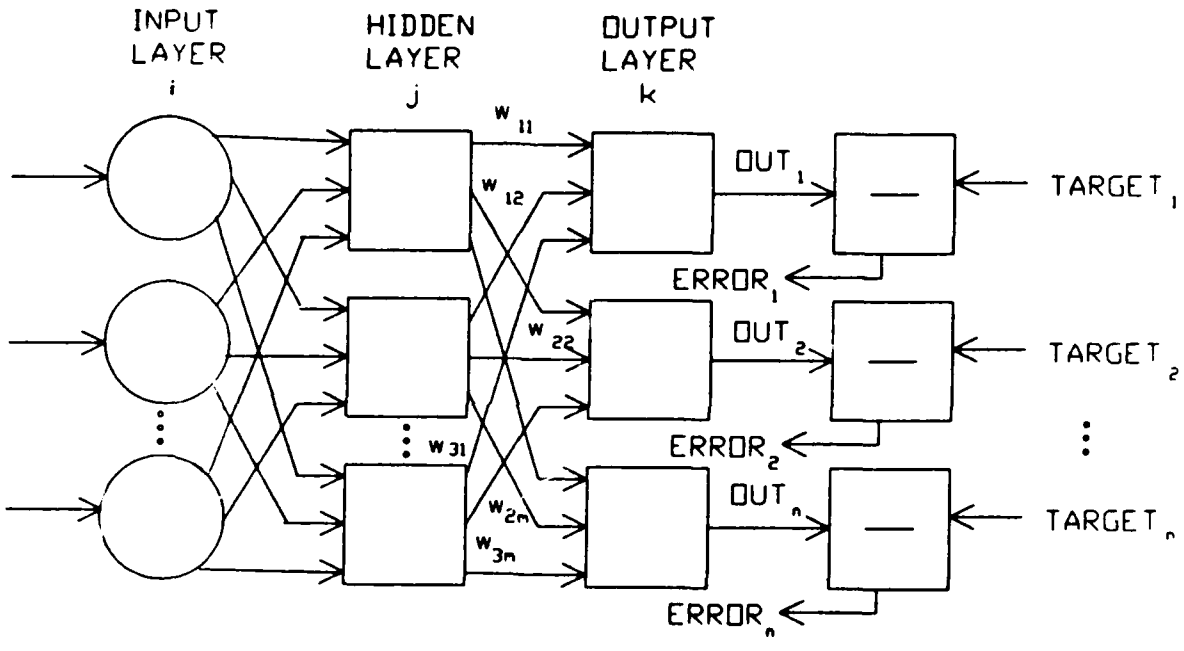


Figure 9: Two-layer backpropagation neural network

output of the i^{th} neuron is:

$$o_i = 1/(1 + e^{-\Sigma w_{ij} o_j}).$$

The network is trained by running a number of cases of (input,output) sets through it and adjusting the weights to minimize a the sum of the squares of the differences between the desired result and the computed result. This quadratic function is the so–called *energy*, *cost*, or *objective* function. The weights are adjusted at random using a steepest descents or a conjugate gradient algorithm.[35] After the network is "trained" it can be run on other input vectors to yield ouput vectors that, hopefully, are good approximations to the correct output.

Application to the Problem of Obtaining Cross Sections from Swarm Data

In order to explore the feasibility of using neural networks on this problem we have been working with a commercial neural net simulator called BRAINMAKER.[36] This is one of a number of such programs as can be seen from the list recently compiled by BYTE Magazine (see the Appendix of this document, which was taken from Ref. 37). We wrote a program to generate cross section sets of the form $\sigma(\epsilon) = \sigma_0/\epsilon^p$, where σ_0 and p are chosen from uniform random numbers in $(10^{-17},10^{-14})$ and (0,1) respectively, and then compute for a range of E/N the distribution function $f(\epsilon)$ and the associated drift velocities, v_d, and characteristic energies, D/μ. We then set up a training set for BRAINMAKER that consisted of the sets $\{v_d\}$ and $\{D/\mu\}$ for ten values of E/N and the cross section $\sigma(\epsilon)$ at nine energies from which the swarm data were computed. The input layer of the network then consists of 20 neurons, one for each value of $v_d(E/N)$ or $D/\mu(E/N)$. The output layer comprises nine neurons, one for each cross section point $\sigma(\epsilon_i)$, i=1 to 9. The network has

two hidden layers of 25 neurons each. In summary the network is described as follows:

Layer	Neurons	Weights
1	20	0
2	25	525
3	25	650
4	9	234

The training tolerance was 5 percent meaning that for the network to be acceptable the cost/energy/objective function of the difference between the $\{\sigma(\epsilon)\}$ defined by the values of the output neurons and the "data" given to the network as part of the training pattern had to be less than or equal to 0.05.

Once the network was "trained" we gave it another file of sets $\{v_d\}$ and $\{D/\mu\}$ corresponding to different sets $\{\sigma(\epsilon_i)\}$ computed with random σ_r and p to see what it predicted for the cross sections. These results are shown in Fig. 10 for the three best (out of 11) cases. These illustrate several things. First, the results denoted by the circles are very good. They all diverge at high energy because the highest E/N that I used (3.0 Td) was too small for v_d and D/μ to be adequately sensitive to the high energy part of the cross section. In addition, we have observed that the results for large cross sections are generally better than the results for small cross sections. We think that this is due to a dynamic range problem with BRAINMAKER that we attribute to its being single precision; it was not really designed for scientific number crunching. This problem could, perhaps, be gotten around by using $\log[\sigma(\epsilon)/\sigma_0]$ where σ_0 is a scale cross section equal to, say, 1 Å^2.

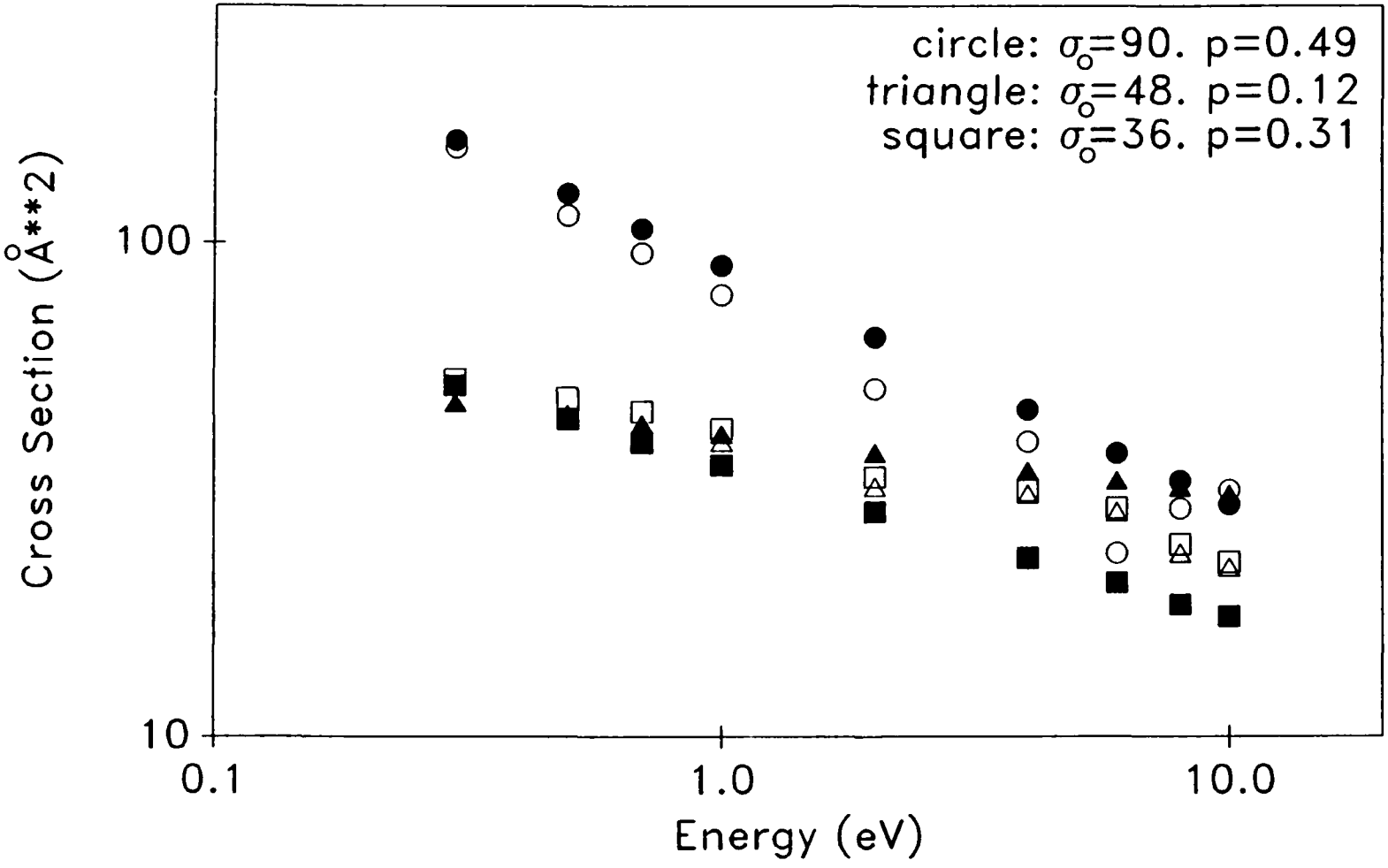


Figure 10: Back-propagation neural network results
Solid symbols: "data", $\sigma(\epsilon) = \sigma_o/\epsilon^p$
Open symbols : neural network output

Application to a Real Gas

In order to test the feasibility of using this kind of neural network to find the pattern in the mapping between $\{v_d(E/N), D/\mu(E/N), \text{etc.}\}$ and $\{\sigma(\epsilon)\}$ we trained BRAINMAKER on 25 sets of $\{v_d(E/N), D/\mu(E/N)\}$ data for cross sections of the form $\sigma(\epsilon) = \sigma_0 \epsilon^p$ where $-1 \leq p \leq +1$. That is, we have some cross sections that increase with energy and some that decrease with energy. We then constructed an input set for Xe with $\{v_d(E/N)\}$ from Hunter, et al.[38] and $\{D/\mu(E/N)\}$ from Koizumi, et al.[39] Unfortunately neither paper presented *both* drift velocity *and* characteristic energy data. This particular network consisted of three layers:

Layer	Neurons	Weights
1	18	0
2	20	380
3	9	189

Fig. 11 displays, so called, Hinton [40] diagrams of the weights of the connections between the neurons of the input layer and the hidden layer and the neurons of the hidden layer and the output layer.

The cross section that the neural network returned in the output layer for Xe in the energy range around the Ramsauer minimum is shown in Fig. 12 along with the $\sigma_m(\epsilon)$ from Hunter, et al,[38] Koizumi, et al,[39] and Frost and Phelps.[18] We see that the neural network gives a respectable estimate of the cross section even though the number of E/N values is small and the energy grid is very coarse.

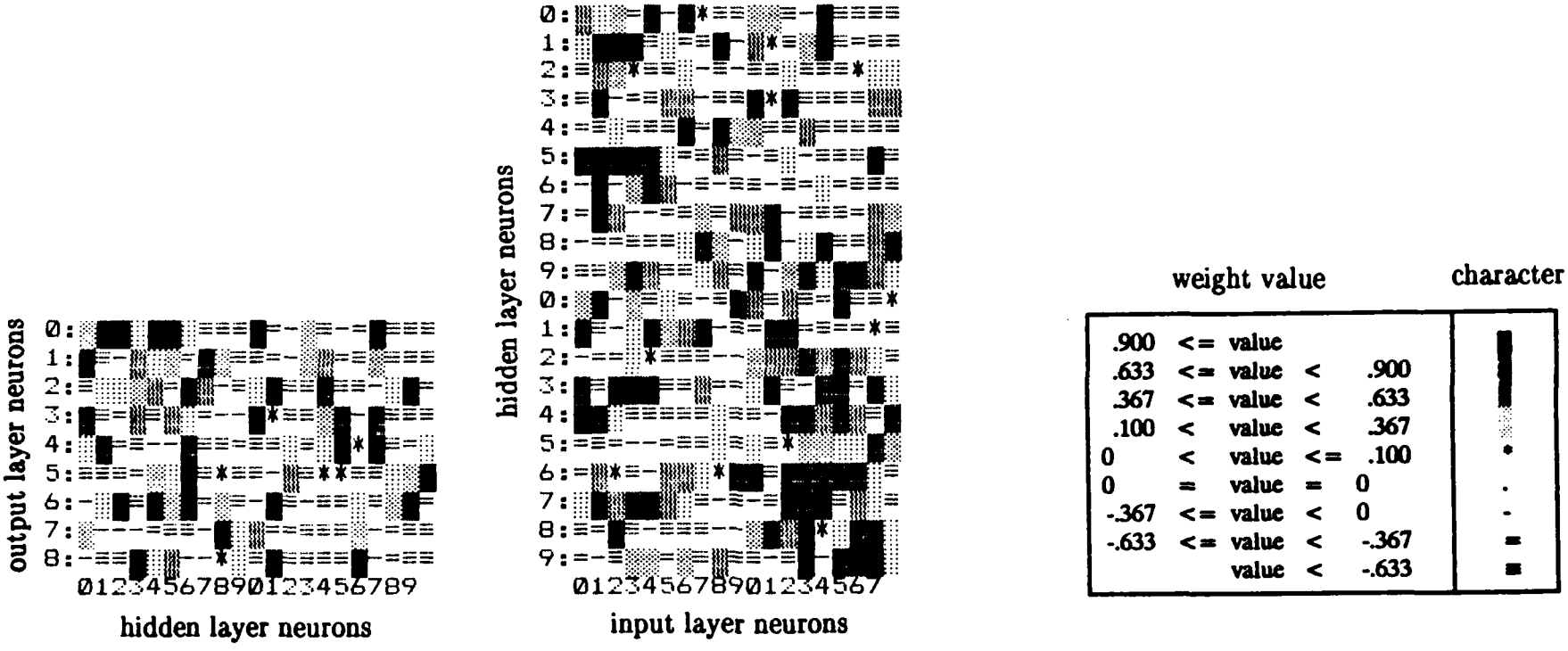


Figure 11: Hinton diagram of weights of connections between neurons

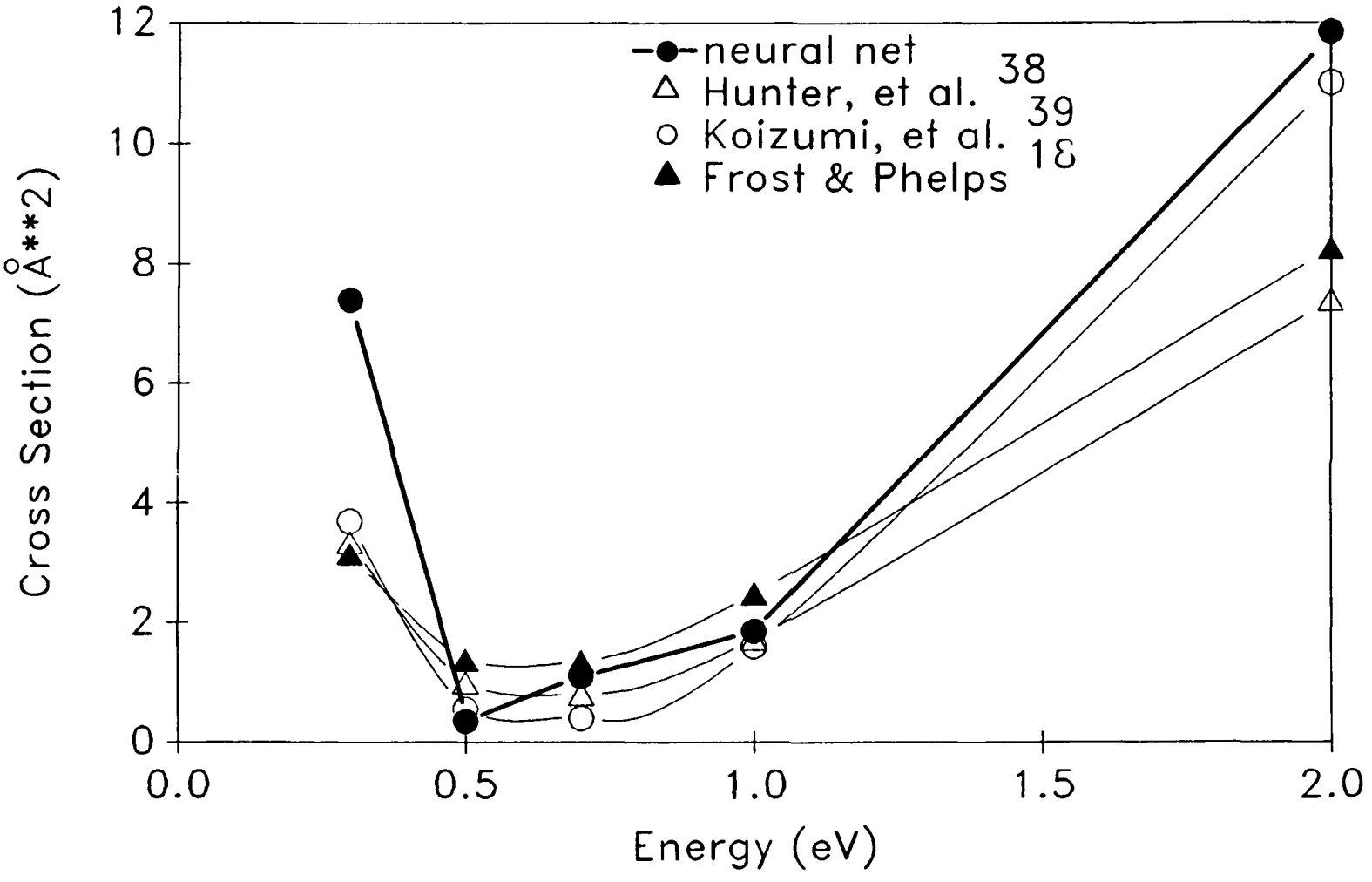


Figure 12: Xe momentum transfer cross section

A Different Neural Network Approach

Jeffrey and Rosner [8,33] have developed a neural network approach to finding the solution to an integral equation that bears further study for application to this problem. This network is a so–called Hopfield [32,34,40–42] or recurrent network. It consists of only one layer, as shown in Fig. 13 (taken from the BRAINMAKER documentation[36]), and is not trained as is the network that we have described above. Rather, it is essentially an iteration algorithm where the output is fed back to modify the input. The mathematical description is developed as follows. We want the solution q to the integral equation $g(y) = \int k(x,y)q(x)dx$, which we write in discrete form as $g_i = \Sigma k_{ij}q_j$. If we write the energy/cost/objective function as a goodness–of–fit function

$$H(\vec{q}) = 1/2\ \Sigma_i\ (g_i^d - g_i)^2$$

and define $I_j = \Sigma k_{ij} g_i^d$ and $T_{ij} = -\Sigma_\alpha\ k_{\alpha i} k_{\alpha j}$ then

$$H(\vec{q}) = -1/2\Sigma_i\Sigma_j\ T_{ij}q_iq_j - \Sigma I_j q_j + 1/2\Sigma_i\ (g_i^d)^2$$

This now is in the standard form investigated by Hopfield where $\vec{q}=(q_1,...,q_N)$ is regarded as the ouput vector of a network of N neurons. By considering $dH/dt=\Sigma(\partial H/\partial q_i)(dq_i/dt)$ Jeffrey and Rosner then show that the update equation for the q_i on the $(n+1)^{th}$ iteration is

$$q_i^{(n+1)} = q_i^{(n)} + \lambda_i[\Sigma_j T_{ij} q_j^{(n)} + I_i]\delta t$$

This gets more complicated for more complicated energy functions. In even the most

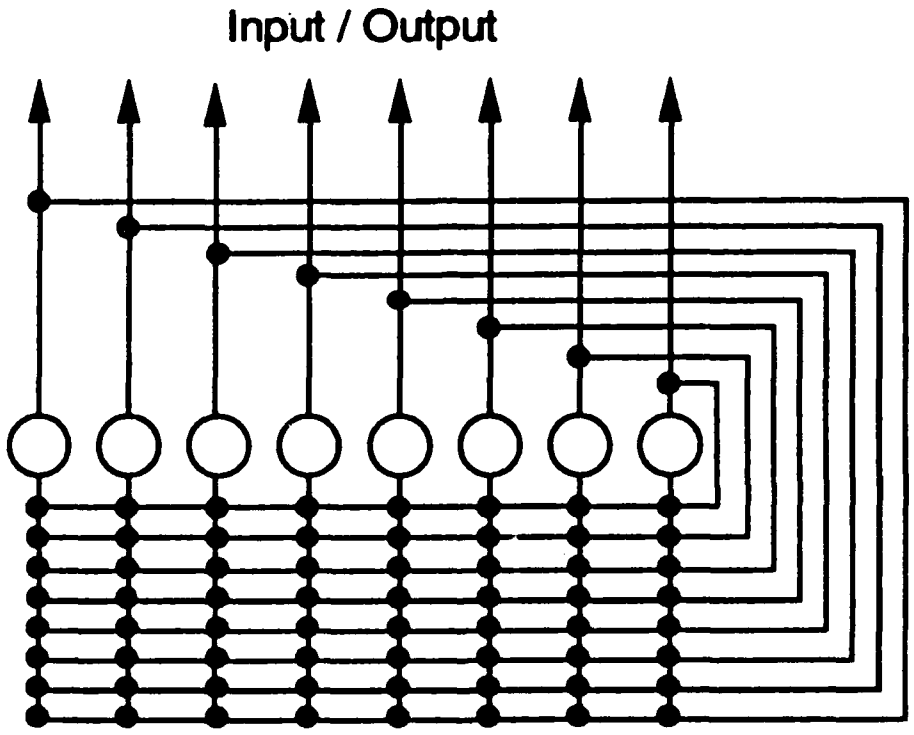


Figure 13: Hopfield single layer recurrent neural network

simple form of our problem (no inelastic processes), the kernel k(x,y) is also a function of q(x). That is,

$$v_d(E/N) \approx \int \{df_o[\epsilon, \sigma_m^{-1}(\epsilon), E/N]/d\epsilon\} \sigma_m^{-1}(\epsilon) \epsilon d\epsilon$$

This does not prevent us from trying this kind of iteration, however. One aspect of this iterative approach that one must watch out for is chaos, as it is known that recurrent networks can be chaotic.[43]

SECTION IV

CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

We have explored three optimization techniques for treating the inverse problem of obtaining electron collision cross sections from electron transport data. These methods were (a) the downhill or creeping simplex; (b) simulated annealing; and (c) neural networks. We devoted the greatest amount of effort to methods (a) and (c). The simplex method was straightforward to implement and, as we saw above, demonstrated a capability for making headway on this problem. Simulated annealing is capable of solving any minimization problem that the simplex can solve and probably much more. It, however, requires substantial further development and may require computational resources beyond what a PC can currently provide. We devoted much effort to investigating the neural network approach because it is very new and has not yet had much application to the problems of applied physics. That approach also has demonstrated some capability in addressing the problem at hand.

The paths for further development of methods (a) and (b) are apparent and have been discussed above. With regard to (a), another possibilty for development is to implement the algorithm developed by N. Karmarkar [44,45] of AT&T Bell Laboratories in 1984. It has been claimed [46] that this algorithm is much faster than the simplex. A perusal of Science Citation Index, however, shows that Karmarkar's algorithm has not yet made it into the physics literature.

We believe that the neural network approach too is worthy of further exploration. The limits of BRAINMAKER to this application have, however, about been exhausted. The next step would be to write a network for this problem with larger numbers of neurons; double precision arithmetic; a capability for having different transfer functions for different layers; allowing different convergence criteria for different energy ranges; and, perhaps, using the Boltzmann training algorithm [31,34,42] (an application of simulated annealing to adjustment of the weights of the connections in the network); i.e., more

flexibility in general. It would be interesting investigate how such a network would perform when trained on a completely artificial data set (probably using more sophisticated functions for the training cross sections than we have used here) as compared to training by feeding it a large set of data on real atoms and molecules. Ultimately we may find that a neural network is good means of getting a rough estimate of a cross section $\sigma(\epsilon)$ that can then be refined using another numerical optimization algorithm. The conventional wisdom has been that neural networks are useful for only very rough solutions and not for accurate scientific calculations but some authors, such as Lapedes and Farber [35] refute that point of view. As this area of research is very much in its infancy, we can expect many new developments in the application and understanding of neural networks in the future.

REFERENCES

1. A.V. Phelps, Rev. Mod. Phys. **40**, 399 (1968).
2. I.P. Shkarofsky, T.W. Johnston, and M.P. Bachynski, **The Particle Kinetics of Plasmas,** (Addison–Wesley, Reading, MA, 1966).
3. L.G.H. Huxley and R.W. Crompton, **The Diffusion and Drift of Electrons in Gases,** (Wiley, New York, 1974).
4. A.V. Phelps in **Swarm Studies and Inelastic Electron–Molecule Collision,** ed. L.C. Pitchford, B.V. McKoy, A. Chutjian, and S. Trajmar (Springer–Verlag, New York, 1987).
5. M.J. Kushner, J. Appl. Phys. **63**, 2532 (1988).
6. W.H. Press,B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, **Numerical Receipes**, (Cambridge Univ. Press, Cambridge, 1986).
7. W. Jeffrey and R. Rosner, Ap. J. **310**, 463 (1986).
8. W. Jeffrey and R. Rosner, Ap. J. **310**, 473 (1986).
9. A.E. Carlsson, Phys. Rev. Letts. **59**, 1108 (1987).
10. S.L. Marple, Jr., **Digital Spectral Analysis**, (Prentice–Hall, Englewood Cliffs, NJ, 1987).
11. B.M. Smirnov, **Physics of Weakly Ionized Gases**, (MIR, Moscow, 1981).
12. J.A. Nelder and R. Mead, Computer Journal 7, 308 (1965).
13. R.W. Crompton, M.T. Elford, and R.L. Jory, Aust. J. Phys. **20** 369 (1967).
14. M.T. Elford, personal communication to Huxley and Crompton (1973).
15. R.W. Crompton, M.T. Elford, and A.G. Robertson, Aust. J. Phys. **23**, 667 (1970).
16. A.G. Robertson, personal communication to Huxley and Crompton (1972).
17. R.W. Warren and J.H. Parker, Phys. Rev. **128**, 2661 (1962).
18. L.S. Frost and A.V. Phelps, Phys. Rev. **136,** A1538 (1964).

19. H.B. Milloy, R.W. Crompton, J.A. Rees, and A.G. Robertson, Aust. J. Phys. **30**, 61 (1977).

20. G.N. Haddad, Aust. J. Phys. **38**, 677 (1985).

21. W.J. Pollock, Trans. Faraday Soc. **64**, 2919 (1968).

22. L.C. Pitchford, S.V. O'Neil, and J.R. Rumble, Phys. Rev. A**23**, 294 (1981).

23. C.W. Duncan and I.C. Walker, J. Chem. Soc. Faraday Trans. 2 **68**, 1514 (1972).

24. J. Ferch, G. Granitza, and W. Raith, J. Phys. B **18**, L445 (1985).

25. D.K. Davies, L.E. Kline, and W.E. Bies, J. Appl. Phys. **65**, 3311 (1989).

26. S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, Science **220**, 671 (1983).

27. S. Kirkpatrick, J. Stat. Phys. **34**, 975 (1984).

28. G. Bhanot, Rep. Prog. Phys. **51**, 429 (1988).

29. **Neural Networks for Computing**, ed. J. Denker (Amer. Inst. of Physics, New York, 1986).

30. D.E. Rumelhart and J.L. McClelland, **Parallel Distributed Processing**, V. 1 (MIT Press, Cambridge, 1986).

31. P. D. Wasserman, **Neural Computing** (Von Nostrand Reinhold, New York, 1989).

32. J. Hopfield and D. Tank, Biol. Cybern. **52**, 141 (1985).

33. W. Jeffrey and R. Rosner, in **Neural Networks for Computing**, ed. J. S. Denker (Amer. Inst. of Physics, New York, 1986).

34. E. Aarts and J. Korst, **Simulated Annealing and Boltzmann Machines** (Wiley, New York, 1989).

35. A. Lapedes and R. Farber, in **Neural Information Processing Systems**, ed. D.Z. Anderson (Amer. Inst. of Physics, New York, 1988).

36. California Scientific Software, Sierra Madre, CA 90124.

37. BYTE Magazine, August 1989, p. 244–245.

38. S.R. Hunter, J.G. Carter, and L.G. Christophorou, Phys. Rev. A **38**, 5539 (1988).

39. T. Koizumi, E. Shirakawa, and I. Ogawa, J. Phys. B **19**, 2331 (1986).

40. G. E. Hinton and T. J. Sejnowski, in **Parallel Distributed Processing**, V. 1, ed. D.E. Rumelhart and J. L. McClelland (MIT Press, Cambridge, 1986).

41. J. J. Hopfield, Proc. Nat. Acad. Sci. **81**, 3088 (1984).

42 J. J. Hopfield and D. W. Tank, Science **233**, 625 (1986).

43. J. M. T. Thomson and H. B. Stewart, **Nonlinear Dynamics and Chaos** (Wiley, New York, 1986).

44. N. Karmarkar, Combinatorica **4**, 373 (1984).

45. A.M. Rockett and J.C. Stevenson, BYTE Magazine (September 1987).

46. G. Kolata, Science **225**, 1379 (1984).

APPENDIX

NEURAL NETWORKS: THEORY AND PRACTICE (BYTE Magazine, August 1989)

For most of their existence, neural networks and neural-network simulations have been solely objects of university-based research. In the last few years, however, researchers and others have founded companies dedicated to producing commercial products based on neural-network technology. To reflect both the academic and commercial aspects of the technology, this resource guide consists of two parts. The In Theory section lists books and articles you can read to learn more about neural networks. The In Practice section lists some of the available neural-network hardware and software products, listed alphabetically by company name.

IN THEORY

Anderson, J. A., M. T. Gately, P. A. Penz, and D. R. Collins. "Radar Signal Categorization Using a Neural Network." *Neural Networks*, 1 Supp. 1, 1988.

Anderson, J. A., and E. Rosenfeld, eds. *Neurocomputing: Foundations of Research*. Cambridge, MA: MIT Press, 1988.

Anderson, J. A., J. W. Silverstein, S. A. Ritz, and R. S. Jones. "Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model." *Psychological Review*, vol. 84, 1977.

Brugge, J. F., and R. A. Reale. "Auditory Cortex." In A. Peters and E. G. Jones, *Cerebral Cortex*. Vol. 4, *Association and Auditory Cortices*. New York, NY: Plenum Press, 1985.

Davis, P. J., and J. A. Anderson. "Nonanalytic Aspects of Mathematics and Their Implication for Research and Education." *SIAM Review*, vol. 21, 1979.

Hadamard, J. *Psychology of Invention in the Mathematical Field*. New York, NY: Dover, 1945.

Hinton, G. E., and J. A. Anderson, eds. *Parallel Models of Associative Memory*, rev. ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1989.

Jones, W. P., and J. Hoskins. "Back-Propagation." BYTE, October 1987.

Josin, G. "Neural Network Heuristics." BYTE, October 1987.

Knapp, A., and J. A. Anderson. "A Theory of Categorization Based on Distributed Memory Storage." *Journal of Experimental Psychology: Learning, Memory and Cognition*, vol. 9, 1984.

Knudsen, E. I., S. du Lac, and S. D. Esterly. "Computational Maps in the Brain." *Annual Review of Neuroscience, Volume 10*. Palo Alto, CA: Annual Reviews, 1987.

Kohonen, T. *Associative Memory*. Berlin: Springer-Verlag, 1977.

Kohonen, T. *Self-Organization and Associative Memory*. Berlin: Springer-Verlag, 1987.

Kosko, B. "Constructing an Associative Memory." BYTE, September 1987.

Lang, K. J., and M. J. Witbrock. "Learning to Tell Two Spirals Apart." In *Proceedings of the 1988 Connectionist Models Summer School*, D. S. Touretzky, G. E. Hinton, and T. J. Sejnowski, eds. San Mateo, CA: Morgan Kaufmann Publishers, 1988.

Lippmann, R. "An Introduction to Computing with Neural Nets." *IEEE ASSP Magazine*, vol. 4, April 1987.

McClelland, J. L., and D. E. Rumelhart, eds. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vols. 1 and 2. Cambridge, MA: MIT Press, 1986.

McClelland, J. L., and D. E. Rumelhart. *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs and Exercises*. Cambridge, MA: MIT Press, 1988.

McCulloch, W. S. *Embodiments of Mind*. Cambridge, MA: MIT Press, 1988.

Mead, C. *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1988.

Minsky, M., and S. Papert. *Perceptrons*. Cambridge, MA: MIT Press, 1969 and 1988.

O'Shaughnessy, D. *Speech Communications*. Reading, MA: Addison-Wesley, 1986.

Pomerleau, D. A. "ALVINN: An Autonomous Land Vehicle in a Neural Network." In *Advances in Neural Information Processing Systems 1*, D. S. Touretzky, ed. San Mateo, CA: Morgan Kaufmann Publishers, 1989.

Rabiner, L. R., and R. W. Schafer. *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

Thorpe, C., M. Herbert, T. Kanade, S. Shafer, and the members of the Strategic Computing Vision Lab. "Vision and Navigation for the Carnegie-Mellon NAVLAB." *Annual Review of Computer Science, Volume II*, Joseph Traub, ed. Palo Alto, CA: Annual Reviews, Inc., 1987.

Viscuso, S. R., J. A. Anderson, and K. T. Spoehr. "Representing Simple Arithmetic in Neural Networks." In *Advanced Cognitive Science: Theory and Applications*, G. Tiberghien, ed. London: Horwoods, 1989.

Waibel, A. "Modular Construction of Time Delay Neural Networks for Speech Recognition." *Neural Computation*, vol. 1, March 1989.

Waibel, A., H. Sawai, and K. Shikano. "Modularity and Scaling for Phonemic Neural Networks." In *IEEE Transactions on Acoustics Speech and Signal Processing* (in press).

Wasserman, P. *Neural Computing, Theory and Practice*. New York, NY: Van Nostrand Reinhold, 1989.

IN PRACTICE

The Brain Simulator $99
Runs under MS-DOS
Tutorial software for neural circuit design
Abbot, Foster & Hauserman
44 Montgomery, Fifth Floor
San Francisco, CA 94014
(800) 562-0025
(415) 955-2711
Inquiry 1181.

N-NET
MS-DOS version $895
VAX/VMS version .... starting at $2995
Integrated neural-network development system; uses functional link net architecture
AI Ware, Inc.
11000 Cedar Ave., Suite 212
Cleveland, OH 44106
(216) 421-2380
Inquiry 1182.

BrainMaker $99.95
Runs under MS-DOS
Neural-network simulation software; supports five types of nodes and can process up 500,000 connections per second
California Scientific Software
160 East Montecito, Suite E
Sierra Madre, CA 91204
(818) 355-1094
Inquiry 1183.

Cognitron
MS-DOS Windows and Mac versions $600
INMOS transputer version $1800
Neural-network/parallel-processing prototyping and delivery system
Cognitive Software, Inc.
703 East 30th St.
Indianapolis, IN 46205
(317) 924-9988
Inquiry 1184.

Connections $87.95
Runs under MS-DOS
A traveling-salesman demo modeled after Hopfield networks

NetWurkz $79.95
Runs under MS-DOS
Elementary introduction to neural networks
DAIR Computer Systems
3440 Kenneth Dr.
Palo Alto, CA 94303
(415) 494-7081
Inquiry 1185.

Savvy Text Retrieval System
Savvy Signal Recognition System
Savvy Vision Recognition System
(Call for pricing)
Run under VAX/VMS, MS-DOS, and Unix
Libraries of C subroutines that use neural technology to solve real-world problems
Excalibur Technologies
2300 Buena Vista SE
Albuquerque, NM 87106
(505) 764-0081
Inquiry 1186.

ANZA ..... from $7000
AT-compatible neural-network coprocessors; includes software and programming interface
ANZA Plus ..... from $12,500
AT-compatible neural-network coprocessors
ANZA Plus/VME $24,950
Neural-network coprocessor for Sun workstations
AXON $1950
A neural-network description language
Neural Network Development Toolkit $3950
For ANZA Plus and ANZA Plus/VME systems
Ports C programs into ANZA Plus and ANZA Plus/VME formats; includes AXON
ExploreNet
MS-DOS version $995
Sun version $3950
Stand-alone neural-network software
HNC, Inc.
5501 Oberlin Dr.
San Diego, CA 92121
(619) 546-8877
Inquiry 1187.

MD/210 Fuzzy Set Comparator $38
Hardware implementation of Hopfield neurons
Micro Devices
5695B Beggs Rd.
Orlando, FL 32810
(407) 299-0211
Inquiry 1188.

N1000
Neural-network development tools for signal and image processing applications
Including 80386 computer ..... from $19,000
Support package only (for Sun-4, Sun-3, IBM PC AT, and PS/2 Model 50 ..... from $7955

N500 ..... from $495
Runs on IBM PC AT and PS/2 Model 50
Single-unit RCE network software
Nestor, Inc.
1 Richmond Sq.
Providence, RI 02906
(401) 331-9640
Inquiry 1189.

Awareness $275
Runs under MS-DOS
Introduction to four types of neural-network paradigms
Genesis $1095
Runs under MS-DOS
Neural-network development environment
Neural Systems
2827 West 43rd Ave.
Vancouver, BC Canada V6N 3H9
(604) 263-3667
Inquiry 1190.

NeuralWorks Explorer $299
Runs under MS-DOS
An introduction and tutorial on neural networks
NeuralWorks Professional II
MS-DOS and Macintosh versions $1495
Sun-3, Sun-4, and Sun386i versions $2995
NeXT and INMOS transputer versions ..... call for pricing
Neural-network development system
NeuralWorks Designer Pack $1995
MS-DOS and Sun versions
Links Professional II networks with C programs
NeuralWare, Inc.
103 Buckskin Court
Sewickley, PA 15143
(412) 741-5959
Inquiry 1191.

MacBrain $995
Runs on Macintosh
Lets you prototype and deliver neural-network applications
HyperBrain
(Comes with MacBrain)
Toolkit allows you to build neural-network applications within HyperCard
Neurix, Inc.
1 Kendall Sq., Suite 2200
Cambridge, MA 02139
(617) 577-1202
Inquiry 1192.

Owl I, II, III ..... from $349
Libraries of modules for IBM and compatibles that lets you define and access 10 different neural networks
Extension Pack $149
Three additional networks
Olmsted & Watkins
2411 East Valley Pkwy., Suite 294
Escondido, CA 92025
(619) 746-2765
Inquiry 1193.

Intelligent Pattern Recognition Chips $500
Stores a 1000-by-64 matrix of weights and multiplies it with an input vector
Oxford Computer
39 Old Good Hill Rd.
Oxford, CT 06483
(203) 881-0891
Inquiry 1194.

ANSim 2.1 ..... $495
Runs under MS-DOS
13 neural-network models
ANSkit $950
Runs under MS-DOS
Neural-network development system
ANSpec $2995
Runs under MS-DOS
Neural-network specification language
Delta Floating Point Processor $24,950
Runs on IBM PC, AT, PS/2s, and Sun386i
Neural-network accelerator boards
Sigma Neurocomputer Workstations ..... from $31,500
80386-based systems with Delta Processor, ANSkit, Delta C, Delta Macro, and ANSpec
SAIC
10260 Campus Point Dr.
Mail Stop 71
San Diego, CA 92121
(619) 546-6290
Inquiry 1195.

DENDROS-1 ..... $35
Neural-network chip that produces the dot product of the inputs and the connection weights of 22 synapses
DENDROS-1 Evaluation Board ... $695
Uses eight DENDROS-1 chips to create a hardware-based neural network
Syntonics Systems, Inc.
20790 Northwest Quail Hollow Dr.
Portland, OR 97229
(503) 293-8167
Inquiry 1196.

TRW Mark V Neural Processor
Write for pricing information
Runs on VAX/VMS
MC68020-based parallel-processing system includes tools for neural-network applications
TRW
Military Electronics & Avionics Div.
One Rancho Carmel
San Diego, CA 92128
(619) 592-3482
Inquiry 1197.

NeuroShell $195
Runs under MS-DOS
Creates neural-network applications using a modified back-propagation
Ward Systems Group, Inc.
228 West Patrick St.
Frederick, MD 21701
(301) 662-7950
Inquiry 1198.

Inclusion in the resource guide does not indicate that BYTE endorses or recommends either the product or the company. In addition, BYTE accepts no responsibility for any omissions, changes, or errors in the information listed.